# Phoneme-to-Articulatory mapping using bidirectional gated RNN

Théo Biasutto– Lervat, Slim Ouni

▶ **To cite this version:**

Théo Biasutto– Lervat, Slim Ouni. Phoneme-to-Articulatory mapping using bidirectional gated RNN. Interspeech 2018 - 19th Annual Conference of the International Speech Communication Association, Sep 2018, Hyderabad, India. hal-01862587

**HAL Id: hal-01862587**
**https://inria.hal.science/hal-01862587**

Submitted on 27 Aug 2018

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Phoneme-to-Articulatory mapping using bidirectional gated RNN

*Théo Biasutto–Lervat, Slim Ouni*

Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France

theo.biasutto-lervat@loria.fr, slim.ouni@loria.fr

## Abstract

Deriving articulatory dynamics from the acoustic speech signal has been addressed in several speech production studies. In this paper, we investigate whether it is possible to predict articulatory dynamics from phonetic information without having the acoustic speech signal. The input data may be considered as not sufficiently rich acoustically, as probably there is no explicit coarticulation information but we expect that the phonetic sequence provides compact yet rich knowledge. Motivated by the recent success of deep learning techniques used in the acoustic-to-articulatory inversion, we have experimented around the bidirectional gated recurrent neural network architectures. We trained these models with an EMA corpus, and have obtained good performances similar to the state-of-the-art articulatory inversion from LSF features, but using only the phoneme labels and durations.

**Index Terms**: speech production, coarticulation modeling, bidirectional recurrent neural network (BRNN)

## 1. Introduction

Recovering the vocal tract shape from speech acoustics could benefit many automatic speech processing system to enrich for instance the acoustic information for synthesis [1] and recognition [2]. In fact, articulatory features are more robust than acoustic features as articulatory features vary very slowly when compared with speech acoustic features.

Recently, corpora of synchronized acoustic and articulatory data streams, using electromagnetic articulography (EMA) for instance, have become available making possible to apply machine learning models and algorithms like HMM [3, 4] or artificial neural networks [5, 6]. More recently, acoustic-to-articulatory inversion using bidirectional long short-term memory (BLSTM) recurrent neural network (RNN) provides very good performance [7, 8].

In our work, we address the inversion problem differently. One question is whether it is possible to predict articulatory dynamics from the knowledge of the phonetic information only: Does the knowledge of the distribution of the phonemes over time can allow predicting the articulatory dynamics ? So the problem can be seen as a phoneme-to-articulatory inversion.

In the past, this problem has been addressed in a different context, not as an inversion problem, but as a coarticulation modeling problem. There are two standard coarticulation models: (1) the rule-based look-ahead model [9] , and (2) time-locked model [10, 11]. We can also note several studies injecting this phonetic knowledge to help the acoustic-to-articulatory mapping [12, 13, 6].

Motivated by the success of BRNNs in the articulatory inversion task, we have explored in this study the ability of such architecture to generate articulatory dynamics from only the phonetic information, i.e. the phoneme label and their respective duration.

In the following sections, we present the RNN architecture used in this study and a simple yet effective training procedure (respectively section 2 and 3). Then we present our phoneme-to-articulatory mapping experiments (section 4), and finally we present the results that we discuss in the last section.

## 2. Bidirectional Gated RNN

### 2.1. Bidirectional RNN

While feedforward neural networks are universal approximators [14], recurrent neural networks (RNN) have been shown to be turing complete, and thus should be able to approximate any dynamical system [15]. RNNs are able to summarize the input sequence into an internal state using cyclical connection, giving them the ability to learn temporal relationship and correlation between data points.

Formally, a RNN is defined by

$$\begin{aligned} h_t &= \mathcal{H}(x_t, h_{t-1}; \theta) \\ y_t &= W_{output}.h_t + b_{output} \end{aligned} \tag{1}$$

where $h_t$ is the network internal state, $x_t$ the network input and $y_t$ the corresponding output at time $t$. $W_{output}$ is the internal-to-output weight connection, and $b_{output}$ the associated bias vector. $\mathcal{H}$ is the recurrent hidden layer transfer function parametrized by the $\theta$ set.

However, these recurrent networks are limited to the use of *past* information, although knowledge of *future* information could improve the $y_t$ prediction. This statement is particularly true when dealing with speech production, for which it is well established that future phoneme influence the production of the current phone through anticipative coarticulation. Bidirectional RNNs (BRNN) [16] overcomes this limitation using two layers simultaneously trained in positive and negative time direction (see fig. 1).

Extending equation 1 for BRNN gives

$$\begin{aligned} \overleftarrow{h}_t &= \overleftarrow{\mathcal{H}}(x_t, \overleftarrow{h}_{t+1}; \overleftarrow{\theta}) \\ \overrightarrow{h}_t &= \overrightarrow{\mathcal{H}}(x_t, \overrightarrow{h}_{t-1}; \overrightarrow{\theta}) \\ y_t &= W_{output}.\mathcal{M}(\overleftarrow{h}_t, \overrightarrow{h}_t) + b_{output} \end{aligned} \tag{2}$$

where $\mathcal{M}$ is the merge function, usually a simple concatenation but literature also exhibits use of element-wise sum or multiplication. $\leftarrow$ denotes elements related to the backward layer and $\rightarrow$ to the forward layer.

BRNNs have outperformed unidirectional RNNs in several tasks, such as phoneme classification [17], neural machine translation [18] or speech enhancement [19].

### 2.2. Gated RNN

Despite its theoretical abilities, training vanilla RNN to learn long-range dependencies using a gradient descent algorithm is

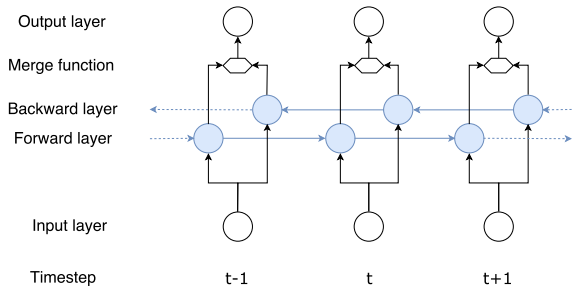


Figure 1: *Bidirectional RNN - blue layers are recurrent and hexagons represent the merge strategy*

still a difficult task due to the *vanishing/exploding* gradient issue [20]. Long Short Term Memory (LSTM) network [21] gets around this issue by computing *increments* to the internal state and so encouraging information to stay for much longer, and by adding to each neural unit three gates which act as weight adjusters in function of inputs and hidden states.

For LSTM, $\mathcal{H}$ is defined by

$$\begin{aligned} f_t &= \sigma(W_f.[h_{t-1}, x_t] + b_f) \\ i_t &= \sigma(W_i.[h_{t-1}, x_t] + b_i) \\ o_t &= \sigma(W_o.[h_{t-1}, x_t] + b_o) \\ C_t &= f_t * C_{t-1} + i_t * \tanh(W_C.[h_{t-1}, x_t] + b_C) \\ h_t &= o_t * \tanh(C_t) \end{aligned} \tag{3}$$

where $C$ is the cell memory and $f$, $i$ and $o$ are respectively the forget, input and output gates. $\sigma$ is the sigmoid function and various $W$ and $b$ correspond to weight connection matrix and bias vector. Concatenation is denoted by $[.,.]$ and $*$ is an element-wise product.

Among several variants of LSTM, the Gated Recurrent Unit (GRU) proposed by Cho et al. [22] has become quite popular and successful. GRU reduces the complexity of LSTM, by removing one gate and the cell memory and so decreasing the number of parameters, which should simplify the training.

For GRU $\mathcal{H}$ is defined by

$$\begin{aligned} z_t &= \sigma(W_z.[h_{t-1}, x_t] + b_z) \\ r_t &= \sigma(W_r.[h_{t-1}, x_t] + b_r) \\ h_t &= z_t * h_{t-1} + (1 - z_t) * \tanh(W_h.[h_{t-1} * r_t, x_t] + b_h) \end{aligned} \tag{4}$$

where $z$ is the update gate and $r$ the reset gate.

Presently, LSTM and variations are well-known for their great performances in language and speech-related tasks for example phoneme classification [23], machine translation [24] or language modeling [25].

# 3. Training procedure

## 3.1. Input and output features

We trained the recurrent networks in a framewise regression scheme where inputs and outputs are synchronized, using a pure stochastic gradient descent. Networks have to predict a sequence of articulator position $\hat{A} = (\hat{a}_0, ..., \hat{a}_T)$ from a phoneme sequence $\Phi = (\phi_0, ..., \phi_T)$, where $\hat{A}$ is as close as possible to the target value $A = (a_0, ..., a_T)$.

The target output $A$ is a sequence of n-dimensional vectors representing the stacked spatial coordinates of each articulator, while the input $\Phi$ is the encoded phoneme sequence. $\phi_t$ is a one-hot vector representing the articulated phoneme at timestep $t$. This encoding preserves the duration of each phoneme without having to explicitly feed this information to the network, and can be seen as a multidimensional binary signal synchronized with the articulator trajectories.

## 3.2. Loss function

As usual in regression task, we used the mean squared error as loss function, and defined the error as the Euclidean distance between $a_t$ and $\hat{a}_t$:

$$\mathcal{L}(A, \hat{A}) = \frac{1}{N} \sum_i \sum_j (a_{ij} - \hat{a}_{ij})^2 \tag{5}$$

with $N$ the sequence size and $a_{ij}$ the j-th dimension of $a_i$.

The partial derivatives of the loss function $\mathcal{L}$ (equation 5) according to the BRNN's parameters were computed with Backpropagation Through Time (BPTT) [26], and the network was fully unfolded for each training sequence.

## 3.3. Adam

The optimization method used to train the network was Adam [27], an adaptive learning rate extension of stochastic gradient descent with many benefits (e.g. appropriate for non-stationary objective and sparse gradients, parameters update invariant to gradient rescaling, intuitive hyper-parameters) and quite popular inside the deep learning community. Adam's authors claim that it combines both the advantages of RMSprop [28] and AdaGrad [29], two other well-known gradient-based optimization algorithms.

$$\begin{aligned} m_t &= \beta_1.m_{t-1} + (1 - \beta_1).g_t \\ \hat{m}_t &= m_t/(1 - \beta_1^t) \\ v_t &= \beta_2.v_{t-1} + (1 - \beta_2).g_t^2 \\ \hat{v}_t &= v_t/(1 - \beta_2^t) \\ \theta_t &= \theta_{t-1} - \lambda.\hat{m}_t/(\sqrt{\hat{v}_t} + \epsilon) \end{aligned} \tag{6}$$

where $\lambda$ is the learning rate, $g_t$ the gradient of the parameter set $\theta_t$ at step $t$, and $\epsilon$ is here for numerical stability. Both $m$ (first moment estimate) and $v$ (second moment estimate) are computed using an exponential moving average (parameters $\beta_1$ and $\beta_2$), moreover a bias correction is applied to reduce the importance of the first samples during the moving average, leading to $\hat{m}$ and $\hat{v}$. In our experiments, we keep the recommended parameters $\lambda = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$.

## 3.4. Early stopping & Learning decay

We used an early stopping strategy to prevent over-fitting and to speed-up the computation [30], which simply consists of stopping the training and rollback to the best model when the validation loss has stopped improving during more than $N_{stop}$ steps. Finally, a learning rate decay was used to slightly improve our final performances. To properly combine our early stopping method and learning rate decay, we reduce the learning rate only when the validation loss stagnates during $N_{decay}$ consecutive epoch, with obviously $N_{decay} < N_{stop}$. The decay is a simple $\lambda_{new} = \lambda * \gamma_{decay}$ with $0 < \gamma_{decay} < 1$. We empirically chose $N_{stop} = 10$, $N_{decay} = 5$ and $\gamma_{decay} = 0.5$ to get

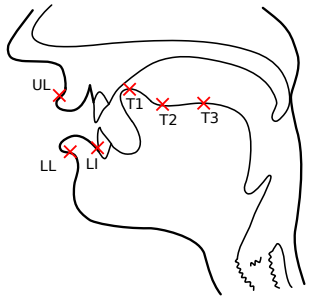


Figure 2: *Sensor coil locations for MNGU0*

Table 1: *Information about MNGU0: number of sentences, duration and number of phonemes for training, testing and validation sets.*

| set | sentences | durations | #phones |
|---|---|---|---|
| training | 1188 | 58'8" | 42,207 |
| testing | 64 | 2'43" | 2,293 |
| validation | 60 | 3'18" | 1,828 |

an acceptable trade-off between performances and computation time.

# 4. Experiments

## 4.1. Training corpus

We obtained the ground-truth trajectories from MNGU0 [31], an articulatory corpus acquired with a Carstens AG500 electromagnetic articulograph [32]. With about one hour of parallel EMA and acoustic data sampled at 200Hz and split among more than 1,300 utterances, MNGU0 is currently the longest and most precise articulatory dataset openly available to our knowledge. All coils are located on the midsagital plane, three along the tongue (on the tip, the body and the tongue dorsum), one on the lower incisive, one on the upper lips and one on the lower lips (fig. 2). This corpus is provided with a phonetic segmentation of all the data. The training phase is performed at each time step on the couple phoneme symbol and articulatory features. When used for inversion, the input of the system is the distribution of the phonetic symbols over time and the output is the articulatory dynamics, as specified in 3.1. For an easier reproducibility of our work and comparison to others methods, we used the training, validation and testing sets proposed by the corpus.

## 4.2. Evaluation metrics

We measured the performances with two well-know metrics, the Root Mean Squared Error (RMSE) and the Pearson correlation $\rho$. For all dimensions of the stacked spatial coordinates $a_i$, both metrics have been computed and averaged over utterances of the test set. Thus we obtain 12 indices by metrics, one for each position of each coil in the midsagittal plan. The final performance of the system is the mean of all averaged indices.

$$
\begin{aligned}
RMSE &= \sqrt{\frac{1}{N}\sum_i (e_i - t_i)^2} \\
\rho &= \frac{\sum_i (e_i - \bar{e})(t_i - \bar{t})}{\sqrt{\sum_i (e_i - \bar{e}) \sum_i (t_i - \bar{t})}}
\end{aligned}
\tag{7}
$$

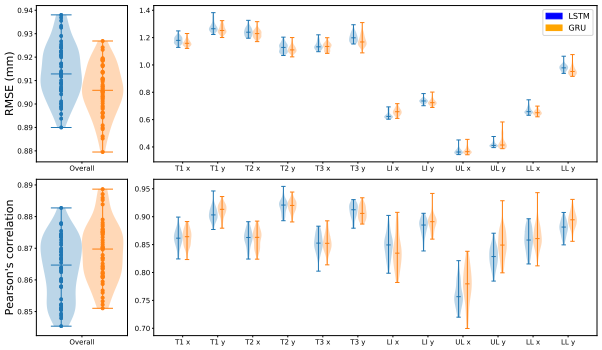


Figure 3: *Violin plot for RMSE (mm) and Pearson's correlation of 50 independent training using either LSTM or GRU units. Subplots on the right correspond to performances per coils, and subplots on the left correspond to the overall performances. Each dot corresponds to a single training.*

where $e_i$ is the prediction at timestep $i$, $t_i$ the associated target, $\bar{e}$ is the mean of the predicted values and $\bar{t}$ the mean of the target values.

## 4.3. Performances: LSTM vs. GRU

To compare LSTM and GRU, we experimentally fixed good enough parameters for the network depth and width, i.e. 2 hidden layers and 256 units by a hidden layer (128 forward, 128 backward). This choice has been motivated by its closeness to the network architecture employed in the studies presented by Liu et al. [8] and by Zhu et al. [7] for state-of-the-art acoustic-to-articulatory inversion. As the training is stochastic and can lead to different results, both LSTM and GRU networks were independently trained and evaluated fifty times using MNGU0.

Figure 3 clearly exhibits highly correlated values and good RMSE for each coil. With an average RMSE of 0.6mm, the jaw (LI) and the lips (UL, LL) dynamics are particularly realistic, comparable to state-of-the-art acoustic-to-articulatory inversion. We even reached excellent result for the upper lip (UL) with RMSE around 0.4mm. The tongue control has a median RMSE per coil around 1.2mm. The RMSE difference between both architectures is 0.007mm for the median, and there is a difference of about 0.01mm between the extrema. We consider that the performance is very good. Figure 5 shows an example of an articulatory trajectory inferred using our method in comparison with ground truth. Globally, the trajectories are very similar to the original ones with very good correlation.

GRU performances are slightly better than LSTM, but the gap is insignificant which seems to corroborate the work of Greff et al. on gated recurrent architecture comparison [33]. A common explanation for this behavior is that for an equivalent number of units, a GRU layer contains fewer parameters, so it should be easier to train and less prone to over-fitting.

## 4.4. Performances: Depth vs. Width

After selecting GRU over LSTM, we explored different network depth and size for this specific unit. Figure 4 exposes the overall performances for networks containing up to 4 layers, where each layer contains 32 to 256 units. Each configuration has been trained twenty times, plain lines represent the median performances, dotted lines the second and third quartile, and triangles the extrema.

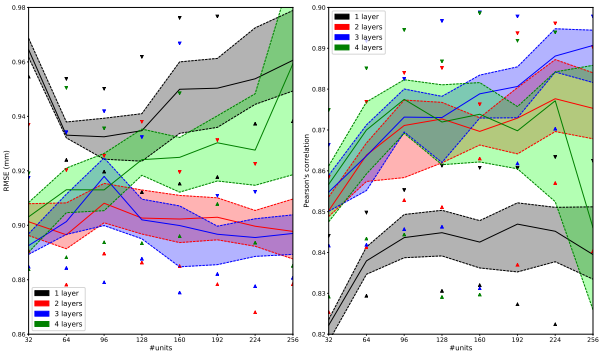


Figure 4: *Overall performances for twenty training when varying the number of units per layer and the number of layers. Plain lines represent the median, dotted lines the second and third quartile, triangles indicate extrema.*

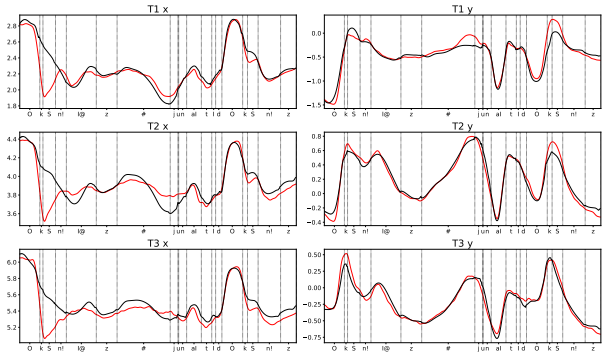


Figure 5: *Tongue trajectory inferred from the sentence "Auctioneers: United Auctions" (mngu0_s1_0220) with the best trained model. Red lines are the ground-truth and black lines are the inference, phonemes are delimited by dotted vertical lines.*

A first observation is the nice performances of the previous network (2 layers and 128 units by a layer) compared to other architecture. This fact consolidates our conviction in the nearness of coarticulation modeling and articulatory inversion tasks, indeed the complexity of both tasks seems to be quite equivalent for a BRNN. Secondly, the single layer models performs quite badly, maybe because the lack of deepness prevents the model to learn the complex relationship between phonetic sequence and articulatory kinematics. Finally, the four-layered models also perform badly, especially when augmenting the number of units by layer. This phenomenon is certainly explained by the huge numbers of parameters, the lack of data, and/or by a training procedure not adapted for such really deep models.

The lowest RMSE is 0.868mm, with 2 layers of 224 units, which is in the same range as the acoustic-to-articulatory inversion results from LSF features using deep mixture density network [5] (0.885mm) or when using an architecture FBB[1] [7] (0.889mm), or FFBB[1] and a more complex training procedure, i.e. greedy layer-wise pretraining with RMSProp followed by a fine-tuning using SGD (0.816mm) [8]. However, it should be noted that acoustic-to-articulatory mapping from MFCC features has the best performance (0.565mm) [7].

## 5. Concluding remarks

In this paper, we explored the use of bidirectional gated recurrent neural networks to map phonetic sequence to an articulatory trajectory. We compared two well-known architectures of gated RNN, LSTM and GRU, and concluded that both methods provide similar performance, with a slight advantage to GRU implementation. We also experimented around the depth of the network and the number of units per layer, and observed that two or three layers seems to be optimum.

Using a simple training procedure, we managed to get performances similar to state-of-the-art acoustic-to-articulatory inversion from LSF features ($0.868mm$ vs. $0.816mm$). This result suggests that phonetic information (i.e. symbol and duration) contains knowledge rich enough to infer articulatory dynamics, even without the explicit coarticulation information present in the acoustic signal. In fact, this richness is embedded in the phoneme information represented by the manner and place of articulation, the phonetic context (the position of the phoneme relatively to the surrounding phonemes), and the duration of each phoneme within this context. This richness seems to model very well the coarticulation phenomena. The success of mapping articulatory dynamics from phonemes suggests that the planning of articulatory movement can be predicted from higher-level information (phonemes) and not necessarily from the lower level information (acoustics).

We are going to investigate further this new manner of driving articulatory dynamics from phonemes, which may make a breakthrough in many fields related to speech production, synthesis and recognition. For instance, it may be interesting to use this technique in audiovisual speech synthesis to animate the tongue from text. Furthermore, we will also consider studying the combination of higher level information with lower-level information to see their impact on predicting articulatory dynamics. We may consider the exploration of different recurrent neural network architectures and training procedures, and assess their repercussion on the quality of the articulatory dynamics.

[1]F stands for feed-forward and B for bidirectional LSTM, so FBB is a network with one feed-forward layer followed by two layers of BLSTM

## 6. Acknowledgements

This work was supported by PIA2 E-Fran, within the METAL project. Authors would like to thank Korin Richmond for making MNGU0 corpus available and for his advices, and the anonymous reviewers for theirs useful comments.

## 7. References

[1] Z. H. Ling, K. Richmond, J. Yamagishi, and R. H. Wang, "Integrating articulatory features into hmm-based parametric speech synthesis," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 6, pp. 1171–1185, Aug 2009.

[2] S. King, J. Frankel, K. Livescu, E. McDermott, K. Richmond, and M. Wester, "Speech production knowledge in automatic speech recognition," *The Journal of the Acoustical Society of America*, vol. 121, no. 2, pp. 723–742, 2007.

[3] S. Hiroya and M. Honda, "Estimation of articulatory movements from speech acoustics using an HMM-based speech production model," *IEEE Transactions on Speech and Audio Processing*, vol. 12, no. 2, pp. 175–185, Mar. 2004.

[4] L. Zhang and S. Renals, "Acoustic-Articulatory Modeling With the Trajectory HMM," *IEEE Signal Processing Letters*, vol. 15, pp. 245–248, 2008.

[5] B. Uria, I. Murray, S. Renals, and K. Richmond, "Deep architectures for articulatory inversion," in *INTERSPEECH 2012*, 2012, pp. 867–870.

[6] X. Xie, X. Liu, and L. Wang, "Deep Neural Network Based Acoustic-to-Articulatory Inversion Using Phone Sequence Information," in *INTERSPEECH 2016*, Sep. 2016.

[7] P. Zhu, L. Xie, and Y. Chen, "Articulatory movement prediction using deep bidirectional long short-term memory based recurrent neural networks and word/phone embeddings," in *INTERSPEECH 2015, 16th Annual Conference of the International Speech Communication Association, Dresden, Germany, September 6-10, 2015*, 2015, pp. 2192–2196.

[8] P. Liu, Q. Yu, Z. Wu, S. Kang, H. Meng, and L. Cai, "A deep recurrent approach for acoustic-to-articulatory inversion," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2015, pp. 4450–4454.

[9] S. E. G. Öhman, "Numerical model of coarticulation," *The Journal of the Acoustical Society of America*, vol. 41, pp. 310–320, 1967.

[10] A. Löfqvist, "Speech as audible gestures," in *Speech Production and Speech Modeling*, 01 1990, pp. 289–322.

[11] M. M. Cohen and D. W. Massaro, "Modeling coarticulation in synthetic visual speech," in *Models and Techniques in Computer Animation*. Springer Japan, 1993, pp. 139–156.

[12] B. Potard, Y. Laprie, and S. Ouni, "Incorporation of phonetic constraints in acoustic-to-articulatory inversion," *Journal of the Acoustical Society of America*, vol. 123, no. 4, pp. 2310–2323, 2008.

[13] A. Ben Youssef, P. Badin, G. Bailly, and P. Heracleous, "Acoustic-to-articulatory inversion using speech recognition and trajectory formation based on phoneme hidden Markov models," in *10th Annual Conference of the International Speech Communication Association (Interspeech 2009)*, Sep. 2009, pp. 2255–2258.

[14] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, pp. 359–366, July 1989.

[15] H. Siegelmann and E. Sontag, "On the computational power of neural nets," *Journal of Computer and System Sciences*, vol. 50, pp. 132–150, February 1995.

[16] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, pp. 2673–2681, November 1997.

[17] A. Graves, S. Fernández, and J. Schmidhuber, "Bidirectional lstm networks for improved phoneme classification and recognition," in *Artificial Neural Networks: Formal Models and Their Applications – ICANN 2005*, 2005, pp. 799–804.

[18] D. Britz, A. Goldie, M.-T. Luong, and Q. Le, "Massive exploration of neural machine translation architectures," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 1442–1451.

[19] M. Wöllmer, Z. Zhang, F. Weninger, B. Schuller, and G. Rigoll, "Feature enhancement by bidirectional lstm networks for conversational speech recognition in highly non-stationary noise," in *International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 6822–6826.

[20] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, pp. 157–166, March 1994.

[21] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, pp. 1735–1780, November 1997.

[22] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Oct. 2014.

[23] A. Graves, S. Fernández, and F. Gomez, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *In Proceedings of the International Conference on Machine Learning, ICML 2006*, 2006, pp. 369–376.

[24] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proceedings of the International Conference on Learning Representations (ICLR)*, September 2015.

[25] W. D. Mulder, S. Bethard, and M.-F. Moens, "A survey on the application of recurrent neural networks to statistical language modeling," *Computer Speech & Language*, vol. 30, pp. 61 – 98, 2015.

[26] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation." MIT Press, March 1986, pp. 318–362.

[27] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, 2015.

[28] T. Tieleman and G. Hinton, "RMSprop Gradient Optimization." [Online]. Available: http://www.cs.toronto.edu/tijmen/csc321/slides/lecture_slides_lec6.pdf

[29] J. Duchi, E. Hazan, and Y. Singer, "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization," *J. Mach. Learn. Res.*, vol. 12, pp. 2121–2159, July 2011.

[30] L. Prechelt, "Early stopping but when?" in *Neural Networks: Tricks of the Trade*. Springer Berlin Heidelberg, 2012, pp. 53–67.

[31] K. Richmond, P. Hoole, and S. King, "Announcing the electromagnetic articulography (day 1) subset of the mngu0 articulatory corpus," in *Interspeech 2011*, January 2011, pp. 1505–1508.

[32] J. S. Perkell, M. H. Cohen, M. A. Svirsky, M. L. Matthies, I. Garabieta, and M. T. Jackson, "Electromagnetic midsagittal articulometer systems for transducing speech articulatory movements," *The Journal of the Acoustical Society of America*, vol. 92, pp. 3078–3096, December 1992.

[33] K. Greff, R. K. Srivastava, J. Koutnk, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A Search Space Odyssey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, pp. 2222 – 2232, Oct. 2017.